**/\*\* 8. Sort the given matrix \*\*/**

Given a n x n matrix. The problem is to sort the given matrix in strict order. Here strict order means that matrix is sorted in a way such that all elements in a row are sorted in increasing order and for row ‘i’, where 1 <= i <= n-1, first element of row ‘i’ is greater than or equal to the last element of row ‘i-1’.  
Examples: 

Input : mat[][] = { {5, 4, 7},

{1, 3, 8},

{2, 9, 6} }

Output : 1 2 3

4 5 6

7 8 9

**/\*\* 9. Find the row with maximum number of 1s \*\*/**

Given a boolean 2D array, where each row is sorted. Find the row with the maximum number of 1s.

**Example:**

Input matrix

0 1 1 1

0 0 1 1

1 1 1 1 // this row has maximum 1s

0 0 0 0

Output: 2

**/\*\* 10. Find median in row wise sorted matrix \*\*/**

We are given a row-wise sorted matrix of size r\*c, we need to find the median of the matrix given. It is assumed that r\*c is always odd.

**Simple Method**: The simplest method to solve this problem is to store all the elements of the given matrix in an array of size r\*c. Then we can either sort the array and find the median element in O(r\*clog(r\*c)) or we can use the approach discussed [here](https://www.geeksforgeeks.org/kth-smallestlargest-element-unsorted-array-set-3-worst-case-linear-time/) to find the median in O(r\*c). Auxiliary space required will be O(r\*c) in both cases.

An **efficient approach** for this problem is to use a [binary search](https://www.geeksforgeeks.org/binary-search/) algorithm. The idea is that for a number to be median there should be exactly (n/2) numbers that are less than this number. So, we try to find the count of numbers less than all the numbers. Below is the step-by-step algorithm for this approach:

***Algorithm****:*

1. *First, we find the minimum and maximum elements in the matrix. The minimum element can be easily found by comparing the first element of each row, and similarly, the maximum element can be found by comparing the last element of each row.*
2. *Then we use binary search on our range of numbers from minimum to maximum, we find the mid of the min and max and get a count of numbers less than or equal to our mid. And accordingly change the min or max.*
3. *For a number to be median, there should be (r\*c)/2 numbers smaller than that number. So for every number, we get the count of numbers less than or equal to that by using upper\_bound() in each row of the matrix, if it is less than the required count, the median must be greater than the selected number, else the median must be less than or equal to the selected number.*

**/\*\* 11. Matrix Multiplication | Recursive \*\*/**

**/\*\* 12. Program to multiply two matrices \*\*/**

**/\*\* 13. Program for scalar multiplication of a matrix \*\*/**

**/\*\***

**\* 14. Program to print Lower triangular and Upper triangular matrix of an array**

**\*\*/**

**/\*\* 15. Find distinct elements common to all rows of a matrix \*\*/**

**/\*\* 16. Print a given matrix in spiral form clockwise \*\*/**

**/\*\* 17. Find maximum element of each row in a matrix \*\*/**

**/\*\* 18. Find unique elements in a matrix \*\*/**

**/\*\* 19. Shift matrix elements row-wise by k \*\*/**

**/\*\* 20. Different Operations on Matrices \*\*/**